

**Amendments to the Claims**

1. (original) A method of parallelizing a database query, comprising:  
dividing a received query on a database table into a number of parallel subqueries, each parallel subquery including a discrete non-overlapping range constraint on a partitioning field of the database table; and  
submitting the parallel subqueries to a database management system in place of the received query.
2. (original) The method of claim 1, wherein the discrete non-overlapping range constraints collectively span the entire range of values in the partitioning field.
3. (original) The method of claim 1, wherein said partitioning field is populated by random numbers.
4. (original) The method of claim 3, wherein said random numbers are distributed substantially uniformly.
5. (original) The method of claim 4, wherein the range constraint comprises a range of values of the random numbers in the partitioning field.
6. (original) The method of claim 5, wherein the range constraint for each individual parallel subquery is based on the number of parallel subqueries and an index number of the individual parallel subquery.
7. (original) The method of claim 1, wherein the database query comprises an SQL statement.
8. (original) The method of claim 1, further comprising:  
extending each record of the database table to include the partitioning field; and  
populating the partitioning field of each record with a random number produced by a random number generator having a substantially uniform distribution.
9. (original) The method of claim 1, further comprising:  
receiving individual results of each parallel subquery; and

separately supplying each of the individual results to subsequent parallel operations.

10. (original) The method of claim 1, wherein the number of parallel subqueries is determined by a method comprising:

setting the number of parallel subqueries based on the received query and a preferred number of database records to be processed by each parallel subquery;

issuing a trial database query having a trial range constraint based on the set number of parallel subqueries, said trial database query returning a trial count of matching database records; and

adjusting the number of parallel subqueries until the trial count falls within a predetermined tolerance factor.

11. (original) A computer programmed to parallelize a database query, comprising:

means to divide a received query on a database table into a number of parallel subqueries, each parallel subquery including a discrete non-overlapping range constraint on a partitioning field of the database table; and

means to submit the parallel subqueries to a database management system in place of the received query.

12. (original) The computer of claim 11, further comprising:

means to extend each record of the database table to include the partitioning field; and

means to populate the partitioning field of each record with a random number produced by a random number generator having a substantially uniform distribution.

13. (original) The computer of claim 11, further comprising:

means to set the number of parallel subqueries based on the received query and a preferred number of database records to be processed by each parallel subquery;

means to issue a trial database query having a trial range constraint based on the set number of parallel subqueries, said trial database query returning a trial count of matching database records; and

means to adjust the number of parallel subqueries until the trial count falls within a predetermined tolerance factor.

14. (original) The computer of claim 11, further comprising:  
    means to receive individual results of each parallel subquery; and  
    means to supply each of the individual results to subsequent parallel operations.
15. (original) A machine-readable medium having stored thereon a plurality of instructions for parallelizing a database query, the plurality of instructions comprising instructions to:  
    divide a received query on a database table into a number of parallel subqueries, each parallel subquery including a discrete non-overlapping range constraint on a partitioning field of the database table; and  
    submit the parallel subqueries to a database management system in place of the received query.
16. (original) The machine-readable medium of claim 15, further comprising instructions to:  
    extend each record of the database table to include the partitioning field; and  
    populate the partitioning field of each record with a random number produced by a random number generator having a substantially uniform distribution.
17. (original) The machine-readable medium of claim 15, wherein the range constraint for each individual parallel subquery is based on the number of parallel subqueries and an index number of the individual parallel subquery.
18. (original) The machine-readable medium of claim 15, further comprising instructions to:  
    set the number of parallel subqueries based on the received query and a preferred number of database records to be processed by each parallel subquery;  
    issue a trial database query having a trial range constraint based on the set number of parallel subqueries, said trial database query returning a trial count of matching database records;  
    and  
    adjust the number of parallel subqueries until the trial count falls within a predetermined tolerance factor.
19. (original) The machine-readable medium of claim 15, further comprising instructions to:  
    individually supply the results of each parallel subquery to a subsequent discrete parallel database processing step.

20. (original) A computer system, including:

- a processor coupled to a network;
- an electronic file storage device coupled to the processor; and
- a memory coupled to the processor, the memory containing a plurality of executable instructions to implement a method of parallelizing a database query, the method comprising:
  - dividing a received query on a database table into a number of parallel subqueries, each parallel subquery including a discrete non-overlapping range constraint on a partitioning field of the database table; and
  - submitting the parallel subqueries to a database management system in place of the received query.

21. (original) The system of claim 20, wherein the discrete non-overlapping range constraints collectively span the entire range of values in the partitioning field.

22. (original) The system of claim 20, wherein said partitioning field is a field populated by random numbers.

23. (original) The system of claim 22, wherein said random numbers are distributed substantially uniformly.

24. (original) The system of claim 23, wherein the range constraint comprises a range of values of the random numbers in the partitioning field.

25. (original) The system of claim 24, wherein the range constraint for each individual parallel subquery is based on the number of parallel subqueries and an index number of the individual parallel subquery.

26. (original) The system of claim 20, wherein the database query comprises an SQL statement.

27. (original) The system of claim 20, wherein the method of parallelizing a database query further comprises:

- extending each record of the database table to include the partitioning field; and
- populating the partitioning field of each record with a random number produced by a random number generator having a substantially uniform distribution.

28. (original) The system of claim 20, further comprising:

individually supplying the results of each parallel subquery to a subsequent discrete parallel processing operation.

29. (original) The system of claim 20, wherein the number of parallel subqueries is determined by a method comprising:

setting the number of parallel subqueries based on the received query and a preferred number of database records to be processed by each parallel subquery;

issuing a trial database query having a trial range constraint based on the set number of parallel subqueries, said trial database query returning a trial count of matching database records; and

adjusting the number of parallel subqueries until the trial count falls within a predetermined tolerance factor.

30. (withdrawn) A method of parallelizing a computer processing operation, comprising:

dividing the operation into a number of packages;

separating each package into a query stage and a processing stage, each query stage including a discrete non-overlapping range constraint on a partitioning field of a database;

submitting all of the query stages to a database management system substantially in parallel; and

providing the results of each query stage to its corresponding processing stage.

31. (withdrawn) The method of claim 30, wherein the number of packages is determined by a method comprising:

setting the number of packages based on a preferred number of database records to be processed by each query stage;

issuing a trial database query having a trial range constraint based on the preferred number of database records, said trial database query returning a trial count of matching database records; and

adjusting the number of packages until the trial count falls within a predetermined tolerance factor.